Condor Keyboard & Console idle detection

Giovanni Franzini

Condor Keyboard & Console Idle detection

- Source code: `src/condor_sysapi/idle_time.cpp`
- `calc_idle_time_cpp()` is the main function.
- This one calls a series of sub-functions to obtain idle times in different ways, and compare them in order to have a good assessment:

```
if( sysapi_has_bad_utmp == TRUE)
        all_pty_idle_time()
else
        utmp_pty_idle_time()
```

`dev_idle_time()` (console and mouse Unix files)

`km_idle_time()`

`sysapi_last_x_event`

`utmp_pty_idle_time()`

- Reads `utmp` file to find terminals assigned to users currently logged in, and uses `dev_idle_time()` to obtain idle time of each one (using the associated file descriptor). The function returns the minimum of these times.
- We are using VMs, so every user has a pseudo-terminal which is located in `/dev/pts/` folder (Linux).
- Advantages:
  - Works correctly with any number of users logged in the same VM.
  - Few Condor dependencies (safe_file methods, `dev_idle_time()`)
- Disadvantages:
  - "*elements in the utmp structure may change size based upon if you are compiling in 32 bit or 64 bit mode, but the byte lengths of the fields in the utmp file might not.*"

    Condor programmer 01/09/2002

`all_pty_idle_time()`

- Looks for pseudo-terminal files into the default directories (different for different Unix based OS) and uses `dev_idle_time()` to obtain their idle times (as `utmp_pty_idle_time()` does).
- Not yet analyzed or used.
- Advantages:
  - `utmp` file is not used. Pseudo-terminal files are involved.
- Disadvantages
  - Depends on a lot of Condor files and libraries.

Open points & possible problems

- These methods do not work on every OS. They need modifications in order to be used on different architectures (code must be recompiled). It is necessary to understand what type of OS the VMs will use (Linux, AIX, Solaris, etc.)
- A Condor update might change some of the used functions and libraries. Probably a code review will be necessary.
  - Use Condor source code...
  - … or write new code with similar functions?
  - Maybe in Python (pyutmp 0.2.1, utmp interface) or other language than C++?
  - How much time will need?
- CPU idle time, memory usage statistic (and maybe other useful information for VM idle detection) still missing.